

DAFTAR TABEL

Tabel 1.1 Tabel <i>Confusion Matrix</i>	10
Tabel 3.1 Tabel <i>preprocessing</i> dan augmentasi data.....	15
Tabel 4.1 Rasio pembagian <i>dataset</i>	22
Tabel 4.2 Hasil performa <i>training</i> rasio 50/20/30.....	23
Tabel 4.3 Hasil performa <i>training</i> rasio 60/20/20.....	24
Tabel 4.4 Hasil performa <i>training</i> rasio 80/10/10.....	25
Tabel 4.5 Hasil performa <i>training</i> rasio 70/20/10.....	26
Tabel 4.6 Hasil <i>training</i> skenario pembagian <i>dataset</i>	27
Tabel 4.7 Hasil performa <i>training</i> dengan epoch 150.	28
Tabel 4.8 Hasil performa <i>training</i> dengan epoch 75.	29
Tabel 4.9 Tabel hasil <i>training</i> skenario epoch.	29
Tabel 4.10 Tabel skenario pengujian <i>alpha</i>	30
Tabel 4.11 Tabel hasil skenario pengujian <i>alpha</i>	31
Tabel 4.12 Tabel hasil pengujian latar belakang putih.	32
Tabel 4.13 Tabel hasil pengujian latar belakang bermotif.	33
Tabel 4.15 Tabel hasil pengujian jarak 50cm.....	34
Tabel 4.16 Tabel hasil pengujian jarak 25cm.....	35
Tabel 4.17 Tabel hasil pengujian 10 lux.	37
Tabel 4.18 Tabel hasil pengujian 700 lux.	38
Tabel 4.19 Tabel hasil pengujian 20.000 lux.....	38

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Bahasa merupakan alat utama berkomunikasi dan sudah menjadi faktor penting dalam kehidupan sehari-hari. Selain bahasa yang sudah dikenal secara umum dan dipraktikkan oleh banyak bagian masyarakat, terdapat juga bahasa yang acap digunakan oleh masyarakat-masyarakat yang memiliki disabilitas seperti contohnya bahasa isyarat, namun belum banyak terdapat metode yang memperkenankan masyarakat yang tidak mengerti bahasa isyarat untuk berkomunikasi dengan masyarakat Bisu atau Tuli.

Merujuk pada data Sistem Informasi Manajemen Penyandang Disabilitas dari Kementerian Sosial pada tahun 2019 tentang persentase penyandang disabilitas di Indonesia berdasarkan jenis, Tuli mencakup 7,03% diantaranya atau sekitar 637.535 orang [1].

Berdasar pada keadaan tersebut, diajukanlah sebuah solusi sistem yang dapat membaca bahasa isyarat kemudian mentranslasikannya ke alfabet bahasa indonesia yang dapat dimengerti oleh mayoritas masyarakat. Solusi ini dipilih karena dirasa perlunya ada metode berkomunikasi yang lebih inklusif terhadap mayoritas kelompok masyarakat. Sebelumnya sudah ada penelitian serupa tetapi terbatas pada kemampuan sistem yang hanya bisa dijalankan melalui perangkat komputer, pada penelitian ini akan digunakan YOLOv5 untuk algoritma pembantu dan penerapannya pada aplikasi mobile untuk memudahkan penggunaan sistem. Solusi ini dapat membantu masyarakat untuk belajar bahasa isyarat.

1.2. Rumusan Masalah

Permasalahan yang akan menjadi dasar dari penelitian ini yaitu:

1. Bagaimana cara yang efektif agar dapat membantu masyarakat untuk belajar bahasa isyarat BISIND?
2. Bagaimana mengetahui performa algoritma dengan melakukan pengujian *hyperparameter* pada algoritma, posisi deteksi objek, latar belakang objek, dan tingkat pencahayaan pada objek?

1.3. Tujuan dan Manfaat

Berikut merupakan tujuan yang mendasari dilakukannya penelitian ini yaitu:

1. Membangun sistem deteksi bahasa isyarat BISINDO dengan algoritma YOLOv5.
2. Melakukan pengujian *hyperparameter* pada algoritma, inferensi posisi deteksi objek, dan pencahayaan pada objek.

1.4. Batasan Masalah

Berikut merupakan batasan masalah dari penelitian ini, antara lain:

1. Jenis bahasa isyarat yang akan diteliti adalah sistem Bahasa Isyarat Indonesia (BISINDO).
2. Arsitektur algoritma yang akan digunakan adalah *You Only Look Once V5*.
3. Dataset yang digunakan untuk keperluan pelatihan adalah *dataset* berupa gambar gerakan bahasa isyarat alfabet A hingga Z dari kamus Bahasa Isyarat Indonesia dalam bentuk gambar JPEG.
4. Sistem inferensi pada penelitian ini menggunakan aplikasi yang dijalankan pada perangkat *mobile*.

1.5. Metodologi Penelitian

Dalam pengerjaan Tugas Akhir, digunakan beberapa metode dalam pelaksanaannya yaitu:

1. Bimbingan dengan Dosen Pembimbing

Kegiatan bimbingan dengan dosen pembimbing 1 dan 2 dilakukan secara rutin setiap dua minggu sekali untuk mendiskusikan kemajuan penelitian.

2. Studi Literatur

Pengumpulan referensi dan hasil penelitian-penelitian sebelumnya yang berhubungan dengan bahasa isyarat dan algoritma model.

3. Pengumpulan dan Verifikasi Dataset

Proses pengumpulan dataset yang merupakan gabungan dari data-data yang diambil sendiri dan tersedia di internet berupa 26 alfabet foto tangan manusia berjumlah 42 gambar untuk masing masing alfabetnya. Kemudian hasil pengumpulan dataset diverifikasi oleh Yayasan SLB yang sudah berpengalaman dengan BISINDO mengenai dataset apakah sudah sesuai kaidah kaidah BISINDO. Hasil verifikasi ini dapat dilihat pada Lampiran A.

4. Implementasi *Training* Model

Di tahap ini dilakukan pengimplimentasian algoritma YOLOv5 terhadap dataset-dataset yang sudah dikumpulkan pada tahap sebelumnya.

5. Pengujian Model

Pengujian model meliputi uji distribusi dataset, uji epoch, dan uji preprocessing. Setiap pengujian akan mencari akurasi terbaik dari setiap pengaturan model *training*.

6. Penulisan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan buku tugas akhir sesuai sistematika penulisan sehingga menjadi buku yang runtut dan teratur.

1.6. Sistematika Penulisan Tugas Akhir

Dalam penyusunan laporan Tugas Akhir ini dilakukan penyusunan secara struktural seperti berikut:

BAB I PENDAHULUAN

Berisi tentang latar belakang, tujuan, rumusan masalah, batasan masalah, metodologi penelitian, dan sistematika penulisan tugas akhir.

BAB II TINJAUAN PUSTAKA

Berisi tentang penjelasan teori-teori yang digunakan dalam penelitian mengenai Bahasa Isyarat dan Algoritma yang digunakan.

BAB III PERANCANGAN SISTEM

Berisi tentang gambaran umum sistem, dataset yang digunakan, analisis kebutuhan sistem, perancangan sistem, dan alur kerja sistem.

BAB IV HASIL PERCOBAAN DAN ANALISIS

Berisi tentang hasil pengujian parameter-parameter algoritma, pembagian data, dan analisis dari hasil yang didapatkan

BAB V KESIMPULAN DAN SARAN

Berisi tentang kesimpulan hasil penelitian dan saran aspek-aspek penelitian yang bisa dikembangkan untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

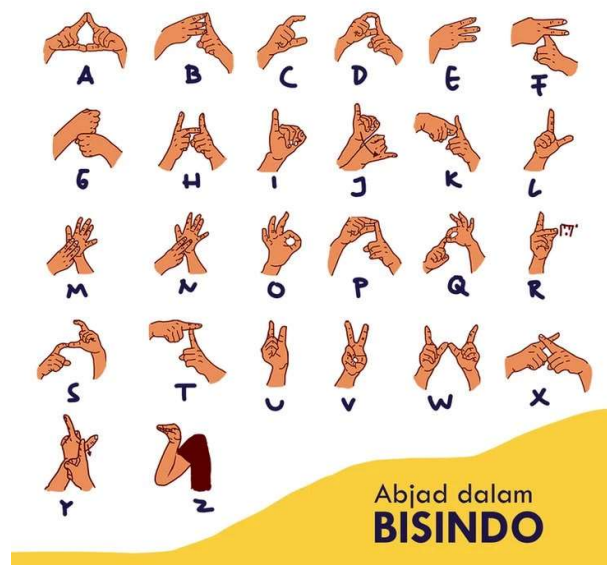
2.1. Bahasa Isyarat

Bahasa Isyarat adalah salah satu metode berkomunikasi dengan orang Tuli. Bahasa Isyarat diungkapkan melalui gerakan-gerakan manual seperti gerakan tangan dan kepala, maupun non-manual seperti pandangan mata dan gestur-gestur arah gerakan tubuh [2].

Selayaknya beragam bahasa yang ada di dunia, bahasa isyarat juga memiliki variasi-variasinya sendiri yang bergantung dengan struktur tatabahasa masing masing negara walaupun kedua negara memiliki bahasa yang sama seperti sistem isyarat *American Sign Language* dan *British Sign Language* yang sama-sama berdasarkan bahasa Inggris tetapi memiliki tatabahasa yang berbeda [3]. Meskipun begitu, tidak berarti semua bahasa isyarat tidak memiliki kesamaan sama sekali, seperti contohnya sistem isyarat Indonesia yang memiliki kesamaan dengan sistem bahasa isyarat Bangladesh.

2.2. Bahasa Isyarat Indonesia

Bahasa Isyarat Indonesia atau BISINDO adalah sistem bahasa isyarat yang umum digunakan oleh Teman Tuli maupun pengguna bahasa isyarat meskipun bahasa yang diakui secara resmi oleh pemerintah adalah Sistem Isyarat Bahasa Indonesia (SIBI). BISINDO lebih umum digunakan karena sistem bahasa syarat ini dikembangkan oleh Teman Tuli itu sendiri [4]. Dari penelitian yang telah dilakukan terkait penggunaan Bahasa Isyarat di Indonesia terhadap 100 orang Teman Tuli, ditemukan bahwa terdapat 31% audiens sangat tidak setuju dengan penggunaan SIBI sebagai bahasa sehari-hari, 37% tidak setuju, 24% cukup setuju, 6% setuju, dan 2% sangat tidak setuju. Sementara pada penelitian yang sama, terdapat 48% audiens yang mendukung sistem BISINDO dalam komunikasi, 43% setuju, 6% cukup setuju, dan 3% tidak setuju [5]. Sistem BISINDO memiliki tatanan sistematis yang diekspresikan dengan isyarat jari, tangan, gestur tubuh, dan gerakan-gerakan yang melambangkan kosakata dalam Bahasa Indonesia. Alfabet-alfabet BISINDO terdiri dari bentuk-bentuk gerakan tangan statis dengan pengecualian alfabet J dan R yang berbentuk gestur.

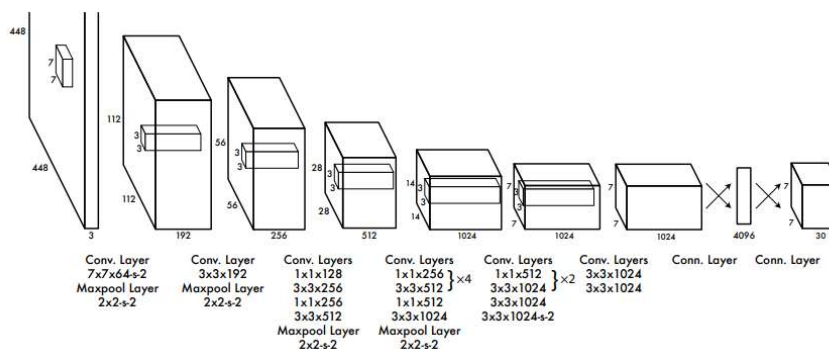


Gambar 2.1 Alfabet dalam sistem Bahasa Isyarat Indonesia [6].

2.3 *You Only Look Once*

You Only Look Once atau biasa disingkat YOLO adalah metode pendeteksian objek yang populer dikarenakan kecepatannya dalam memproses citra. Berbeda dengan CNN dan turunan CNN lainnya, YOLO hanya menggunakan satu lapisan jaringan untuk memprediksi objek apa yang ada dalam gambar atau video yang mengakibatkan lebih tingginya kecepatan YOLO jika dibandingkan dengan CNN dengan konsekuensi akurasi yang sedikit lebih rendah [7].

2.3.1 Arsitektur YOLO



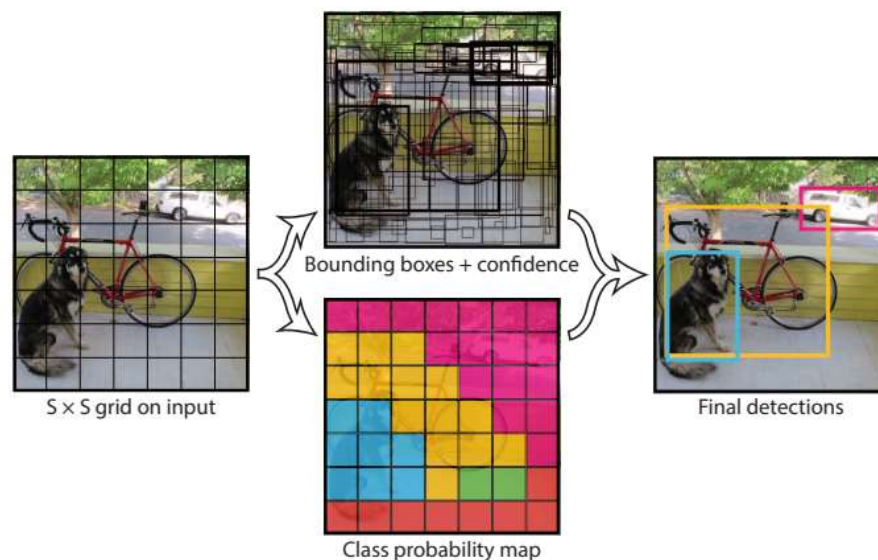
Gambar 2.2 Arsitektur YOLO [8].

Arsitektur YOLO memiliki 24 lapisan konvolusional dan dua lapisan yang terhubung. 24 lapisan konvolusional tersebut bertugas untuk mengambil atribut-

atribut penting dari gambar dan dua lapisan terhubung sisanya bertugas untuk memprediksi kemungkinan keluaran koordinat [9].

2.3.2 Cara Kerja YOLO

Secara garis besar, cara kerja YOLO dibagi lagi menjadi beberapa tingkat. Tingkat pertama yaitu citra yang akan diproses dibagi menjadi beberapa sel jaringan, jumlah sel jaringan dapat dihitung menggunakan cara membagi resolusi gambar dengan jumlah *stride* yang ditentukan agar menghasilkan kotak 8x8, 16x16, dan 32x32. Dikarenakan pada YOLO umumnya digunakan resolusi gambar 640x640, maka *stride* yang digunakan adalah 80x80 untuk objek ukuran kecil, 40x40 untuk objek ukuran sedang, dan 20x20 untuk objek ukuran besar [10].



Gambar 2.3 Rangkaian alur kerja YOLO.

Masing-masing dari kotak pembatas memiliki 4 koordinat dan satu parameter pengukur yaitu x, y, w, h , dan ketepatan. Koordinat (x, y) digunakan untuk menandakan lokasi tengah kotak yang relatif dari letak kotak tersebut. Koordinat (w, h) digunakan untuk menandakan tinggi dan lebar dari gambar secara keseluruhan, sedangkan parameter ketepatan mengukur IoU (*Intersection Over Union*) atau biasa disebut rasio tumpang tindih yang menandakan area tumpang tindih antara kotak objek yang akan dideteksi dan kotak objek kebenaran dasar [11]. Selain 4 koordinat dan satu parameter pengukur tadi, masing masing sel juga memiliki satu parameter lagi yaitu kelas dari objek yang dinotasikan dengan simbol

c , yang hanya ada satu pada setiap sel. Pada YOLO, digunakan metode *Non Maximum Suppression* untuk mengeliminasi sel-sel yang mendeteksi objek yang sama dengan cara melihat objek sel yang memiliki keakuratan paling tinggi dan mengeliminasi sel-sel pembatas lainnya yang mendeteksi objek yang sama [12]. Masing-masing sel pada YOLO hanya dapat memprediksi satu jenis kelas, yang berujung kepada kelemahan metode ini yaitu tidak dapat mengklasifikasikan dua objek berbeda jika mereka berada dalam satu sel yang sama [13].

Masing masing prediksi sel dapat dinotasikan dengan Persamaan 2.1 dimana C adalah jumlah kelas dan B adalah variabel kotak pembatas yang kemudian dikali lima dikarenakan masing-masing kotak memiliki empat koordinat dan satu parameter ketepatan, kemudian dikalikan dengan $S * S$ karena terdapat $S * S$ sel dalam sebuah gambar [14] yang menjadikan persamaan keseluruhan model dapat dinotasikan dengan Persamaan 2.2.

$$p = C + B * 5 \quad (2.1)$$

$$p = S \times S \times (C + B * 5) \quad (2.2)$$

2.4 YOLOv5

YOLOv5 atau biasa disebut juga ultralytics-yolo merupakan versi “informal” dari YOLOv4. Meskipun menggunakan arsitektur yang sama, YOLOv5 berhasil memangkas ukuran model dari yang awalnya sebesar 245 MB menggunakan YOLOv4, menjadi hanya 27 MB dengan YOLOv5 dengan ukuran gambar yang sama. Meskipun begitu, YOLOv5 memiliki AP 36.7%, lebih rendah daripada YOLOv4 yang memiliki AP 41.2% pada konfigurasi *dataset* dan *environment* yang sama [15]. Keunggulan YOLOv5 tidak hanya pada ukurannya saja tetapi juga pada kemudahannya untuk diaplikasikan ke berbagai *platform*.

2.5 Transfer Learning

Transfer learning adalah metode yang menggunakan *pretrained weight* dari model algoritma yang telah dilatih sebelumnya untuk mempersingkat waktu pelatihan dengan tidak mengorbankan akurasi. Informasi tersebut kemudian digunakan lagi untuk melatih model baru dengan parameter, konfigurasi, dan *dataset* yang berbeda [16]. Model YOLOv5 yang digunakan pada penelitian ini adalah YOLOv5s yang sebelumnya dilatih untuk mengenali 80 kelas pada MS

COCO dataset. Untuk mencapai tujuan penelitian, dilakukan *fine tuning* pada model YOLOv5s agar hanya dapat mendeteksi 26 kelas alfabet BISINDO.

2.6 Konfigurasi Variabel

Pada YOLOv5, terdapat beberapa variabel konfigurasi yang digunakan untuk melatih model dalam mencapai hasil yang diinginkan.

2.6.1 Epoch

Epoch adalah istilah yang digunakan untuk menandakan berapa kali algoritma menjalankan satu siklus data *training*. Banyaknya epoch adalah parameter utama yang akan menandakan apakah *dataset* tersebut *overtrained* atau tidak [17].

2.6.2 Pembagian *Dataset*

Pembagian *dataset* berguna untuk mengetahui performa algoritma terhadap berbagai macam skenario data. Hal ini dilakukan untuk mencari di rasio mana algoritma berfungsi paling optimal.

2.7 Parameter Performa

Parameter performa adalah parameter-parameter yang didapatkan setelah melakukan proses training. Nilai-nilai dari parameter ini kemudian akan digunakan untuk menilai apakah hasil training sudah optimal atau belum.

2.7.1 Confusion Matrix

Confusion Matrix adalah tabel yang digunakan untuk menghitung proses sampling dari sebuah model pembelajaran. Tabel ini memiliki 4 kombinasi nilai yaitu TP (True Positive), FP (False Positive), FN (False Negative), dan TN (True Negative). Nilai true positive merupakan nilai ketika hasil prediksi bernilai positif dan sesuai dengan nilai faktual, sedangkan false positive adalah posisi ketika hasil prediksi menghasilkan positif tetapi nilai faktualnya negatif. Sementara itu, false negative merupakan nilai ketika hasil prediksi negatif tetapi nilai faktualnya positif, dan true negative adalah nilai ketika hasil prediksi negatif dan nilai faktualnya juga negatif [18]. Confusion matrix secara umum dapat diilustrasikan dalam tabel pada Tabel 2.1 berikut.

Tabel 2.1 Tabel *Confusion Matrix*

		Nilai faktual	
		+	-
Nilai prediksi	+	True Positive(TP)	False Positive(FP)
	-	False Negative(FN)	True Negative(TN)

2.7.2 Presisi

Presisi adalah rasio prediksi nilai kebenaran dasar positif dengan keseluruhan hasil prediksi positif. Untuk mencari presisi, digunakan persamaan yang didapat dari *confusion matrix* seperti pada Persamaan 2.3.

$$Presisi = \frac{TP}{(TP+FP)} \quad (2.3)$$

2.7.3 Recall

Recall merupakan rasio prediksi positif dengan nilai kebenaran dasar positif. Untuk mencari *recall*, digunakan persamaan yang didapat dari *confusion matrix* seperti pada Persamaan 2.4.

$$Recall = \frac{TP}{(TP+FN)} \quad (2.4)$$

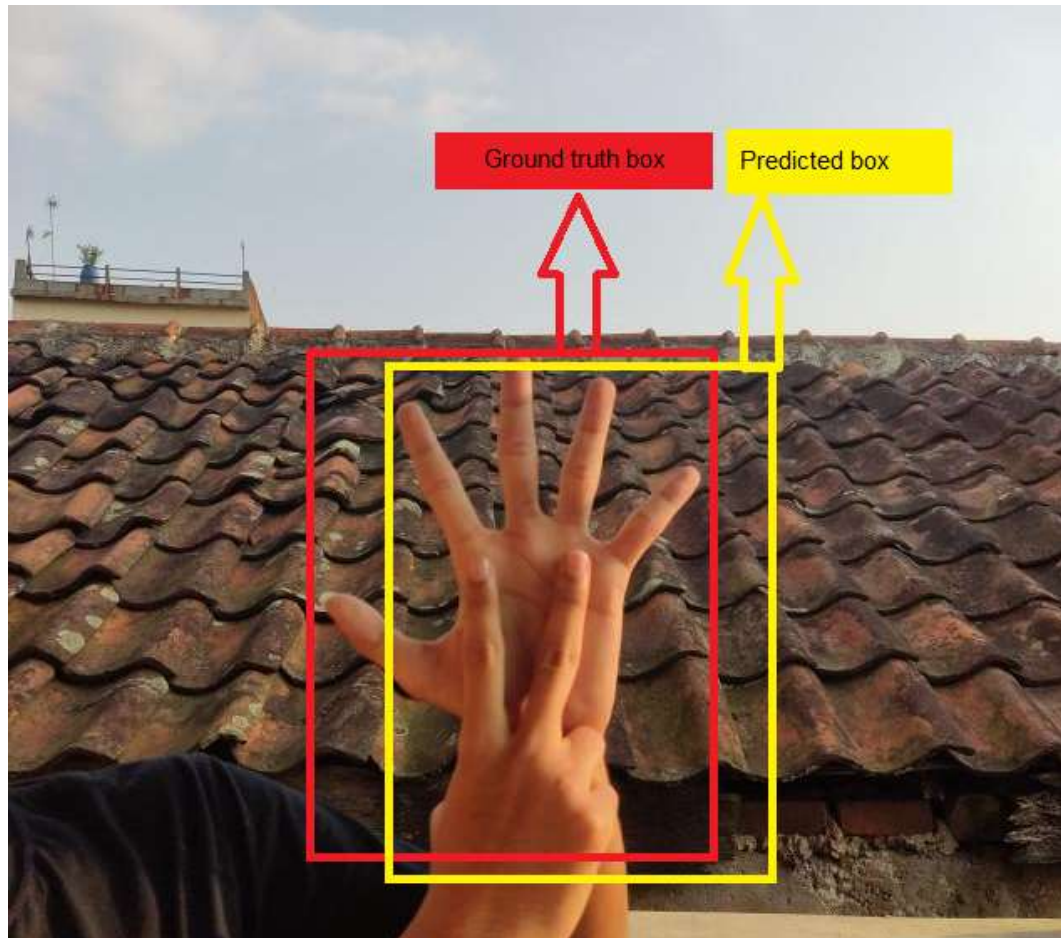
2.7.4 F1 score

F1 *score* adalah nilai yang membandingkan rata-rata dari nilai presisi dan *recall*. Persamaan F1 *score* dapat dilihat pada persamaan 2.5.

$$F1 \text{ score} = 2 \times \frac{Recall \times presisi}{Recall + presisi} \quad (2.5)$$

2.7.5 Intersection Over Union (IoU)

IoU adalah metrik evaluasi keakuratan algoritma dalam mendeteksi objek [19]. IoU membandingkan kebenaran dasar objek dengan *bounding box* yang diprediksi dari model, semakin dekat *bounding box* prediksi dengan *bounding box* kebenaran dasar, maka makin tinggi nilai IoU. IoU memiliki sistem bernama *threshold* yang merupakan sebagai nilai standar atau nilai batas, nilai ini umumnya bernilai 0.5 yang berarti jika objek terdeteksi bernilai >0.5 IoU maka objek akan terdeteksi sebagai *True Positive* dan sebaliknya jika objek terdeteksi bernilai <0.5 IoU maka objek akan terdeteksi sebagai *False Positive*.



Gambar 2.4 Ilustrasi IoU.

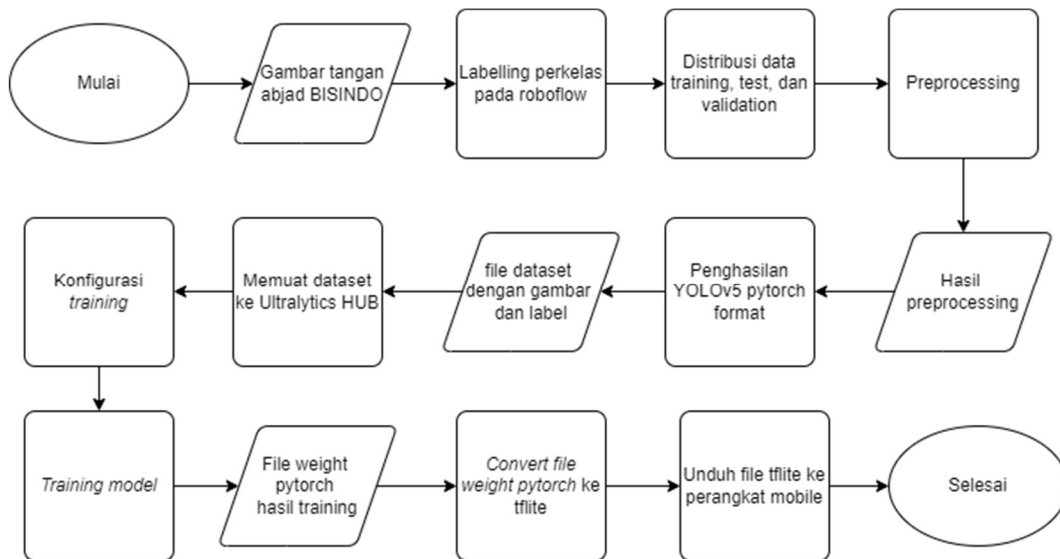
2.7.6 Mean Average Precision

Mean Average Precision atau biasa disingkat mAP adalah matriks yang mengukur performa dari keseluruhan hasil *training*. Dikarenakan pada penelitian ini nilai IoU yang digunakan adalah 0.5, maka model menghasilkan mAP@0.5 IoU. Nilai mAP adalah nilai rata-rata dari *Average Precision*.

BAB III PERANCANGAN SISTEM

3.1. Gambaran Umum Sistem

Sistem yang dirancang untuk sistem berbasis seluler ini terdiri dari 2 bagian yaitu sistem itu sendiri dan pembelajaran mesin. Setelah mendapatkan file *weight* dari hasil *training* algoritma, *weight* tersebut kemudian akan diimpor ke aplikasi *mobile* yang kemudian akan digunakan untuk membaca gerakan bahasa isyarat.



Gambar 3.1 Gambaran umum sistem.

Gambar 3.1 di atas adalah gambaran umum proses pengolahan data menggunakan YOLO yang digunakan dalam perancangan. Setelah mendapatkan *dataset*, perlu dilakukan pelabelan pada masing masing gambar tersebut terlebih dahulu sesuai format YOLO, kemudian setelah selesai diberi label, seluruh gambar akan dibagi menjadi 3 set yaitu *training*, *test*, dan *validation*. Setelah melewati tahap *preprocessing*, data kemudian akan diekstrak kedalam format YOLOv5 Pytorch. Langkah selanjutnya adalah memuat data-data yang sudah dilabel tadi ke dalam Ultralytics HUB untuk dilakukan proses training menggunakan Google Collab. Jika *training* sudah selesai, Ultralytics HUB akan mengubah file *weight* pytorch menjadi tflite dan memuatnya ke aplikasi mobile.